시간 관계성을 기반으로 한 비디오 데이터 모델의 설계 및 구현

최지희[†] · 용환승[††]

요 약

비디오 데이터 자체가 시간적 구조와 공간적 구조로 이루어져 있기 때문에, 비디오 데이터에 대한 내용 기반 검색은 두 관계를 중심으로 이루어 질 수 있다. 본 논문에서는 비디오 데이터 구조가 시간의 흐름에 따라 논리적 계층 구조로 표현 가능하며, 각각의 계층은 각기 시간의 흐름에 따라 시간 관계성을 지닌다는 특성을 반영한 검색 기능을 설계하였다. 그리고 비디오 데이터의 시간적 관계를 계승, 캡슐화, 함수 중복 등의 객체 지향 특성을 이용하여 객체-관계 DBMS로 구현하였다. 기존의 제한적인 시간 함수가 아닌 본 논문에서 제시한 다양한 비디오 데이터의 시간 관계성에 따른 좀 더 확장되고 다양한 시간 함수를 제공함으로써, 사용하기 편리한 인터페이스와, 여러 가지 시간 질의어를 제공한다.

Design and Implementation of the Video Data Model Based on Temporal Relationship

Jihee Choi[†] and Hwan-Seung Yong[††]

ABSTRACT

The key characteristic of video data is its spatial/temporal relationships. In this paper, we propose an content based video retrieval system based on hierarchical data structure for specifying the temporal semantics of video data. In this system, video data's hierarchical structure temporal relationship, inter video object temporal relationship, and moving video object temporal relationship can be represented. We also implemented these video data's temporal relationship into an object-relational database management system using inheritance, encapsulation, function overloading, etc. So more extended and richer temporal functions can be used to support a broad range of temporal queries.

1. 서 론

1.1 연구 배경 및 목적

오늘날 비디오 커뮤니케이션은 우리의 일상생활의 가장 중요한 부분중의 하나이다. TV 프로그램에서부터 비디오 교육에 이르기까지 그 활용범위는 매우 다양하다고 할 수 있다. 과거에는 프로세싱 시간과 저장공간의 제약으로 인해 디지털 비디오 응용 프로그램의 개발이 제한되었다. 그러나 최근에는 많은 용량을 지닌 저장 매체의 개발과 MPEG과 같은 비디오 코딩 표준의 높은 성능으로 인해 디지털 비디오 응용 프로그램이 실용화 되고 있다. 그 중 멀티미디어 데이터베이스는 최근 가장 주목받는 연구분야 중 하나이다. 최근에 등장하는 원격진료, 디지털 도서관, 원격 교육, 여행정보, 분산 CAD/CAM, 지리정보시스템(GIS)등은 일반적인 멀티미디어 데이터베이스 시스템이 될 것이다[3,5].

이러한 미래의 많은 멀티미디어 응용 프로그램들은 이미지와 비디오 데이터의 많은 양을 디지털화 할 것을 요구하고, 탐색, 브라우징, 선택적 재생, 에디팅을 포함하는, 사용자와 상호 작용하는 검색을 요구

† LG Telecom
†† 이화여자대학교 컴퓨터학과 조교수

하게 된다.

결국 이러한 응용 프로그램은 상대적으로 많은 양의 비디오를 접근해야 하기 때문에, 많은 양의 비디오 데이터 중에서 원하는 데이터를 어떻게 하면 빠르고 효율적으로 검색할 것인가 하는 문제를 갖게 된다. 비디오 데이터 자체가 시간적 구조(temporal structure)와 공간적 구조(spatial structure)로 이루어져 있기 때문에, 비디오 데이터에 대한 내용 기반 검색은 두 관계를 중심으로 이루어질 수 있다. 비디오 데이터는 많은 객체들을 포함하고 있으며, 여러 미디어가 혼합되어 있기 때문에, 이들간의 관계성을 이용한 정보검색을 하려면, 시간 동기화, 시간 전후 관계, 공간 관계 등이 먼저 멀티미디어 데이터 모델로써 표현되어져야 한다. 그 중에서 비디오가 다른 이미지 영상과 가장 다른 점은, 시간에 따라 끊임없이 변화한다는 것이다. 따라서 비디오 데이터를 의미적으로 추상화하고, 그들 간의 시간적 표현을 간략히 표현 가능하도록 하는 비디오 모델이 제공되어야 하고, 이를 기반으로 한 시간 질의어(temporal query), 시간 함수(temporal operator)등이 지원되어야 한다.

기존의 비디오 데이터 모델링에 대한 연구를 살펴보면, 의미있는 프레임의 연속을 최하 단위로 규정하고 계층적 구조로 표현하였다. 그러나 새로 등장한 MPEG-4로 인해, 연속된 프레임뿐만 아니라 비디오 데이터내에 존재하는 사람, 집, 차와 같은 의미있는 픽셀의 집합에 대해서도 임의 접근이 가능하게 되었다. 따라서 본 논문에서는 이러한 의미있는 픽셀집합을 비디오 데이터의 최하 단위로 규정하고, 이를 비디오 객체(object)라 부르고자 한다. 최근 들어 이러한 비디오 객체에 대한 연구들이 활발히 진행중이다. 그러나 이러한 연구들은 여러 비디오를 고려한 것이 아니라, 하나의 비디오에서만 정의 가능하고, 대수(algebra)로만 정의하여 질의어로 확장하지 못하였으며, 데이터베이스 구조를 전혀 중요하게 다루지 않았다는 한계점을 지니고 있다[1,2,9]. 또한 기존의 연구에서 보여주는 시간 함수는 두 개의 비디오 데이터가 어떠한 시간 관계를 가지느냐에 따라 True/ False를 리턴해 주는 매우 제한적인 형태였다[11].

즉 지금까지 비디오 데이터의 시간적 구조에 따른 모델링에 대한 연구는 매우 한정된 형태였으며, 제한된 시간 함수로 인하며 다양한 시간 질의어들을 충분히 제공하지 못하였다.

따라서 이러한 단점들을 해결하기 위해, MPEG-4에서 의미하는 비디오 객체까지 고려한 비디오 데이터의 논리적 계층 구조 정의와, 각 계층구조에 따른 시간 관계성을 명확히 하고, 이를 객체 지향 특성을 이용하여, 객체-관계 데이터베이스 시스템(ORDBMS: Object-Relational Database Management System)에 통합함으로써, 편리한 인터페이스와 다양한 시간 질의어의 지원이 필요하다고 할 수 있다.

1.2 연구 내용

본 논문에서는 기존의 비디오 데이터 모델의 제한점과, 시간 함수의 한계점을 보완하기 위해, 시간 관계성을 기반으로 한 비디오 데이터 모델을 설계하고, 이를 객체 지향 특성을 이용하여, 객체-관계 DBMS에 통합하고자 한다.

즉, 비디오 데이터의 구조적 특성을 기반으로, 비디오 데이터의 계층적 구조에 대한 시간 관계성, 비디오내의 각 객체들 간의 시간 관계성, 움직이는 비디오 객체의 시간 관계성을 제시한다. 이러한 비디오 데이터의 시간적 관계를 객체-관계 DBMS에서 제공하는 타입 상속, 리스트와 같은 데이터 타입을 기반으로 데이터 모델을 정의하고, 함수 중복(function overloading)기능을 이용하여 다양한 시간 함수를 지원함으로써, 사용자에게 다양한 시간 질의어와 편리한 인터페이스를 제공한다. 또한 본 논문에서 제안하는 비디오 데이터의 시간 관계성 모델과 시간 함수의 효율성을 검증하기 위해, 객체-관계 DBMS 인 Informix Universal Server[16]와 Informix Web Datablade Module을 사용하여, 웹상에서 비디오 데이터를 시간 관계성에 기반하여 원하는 비디오 데이터를 검색하고 재생 가능하도록 하는 검색 시스템을 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 비디오 데이터 검색 모델을 설계하기 위한 관련 연구분야에 대해 소개한다. 3장에서는 비디오 데이터의 특성을 정의하고, 시간 관계성을 기반으로 비디오 데이터 검색 모델을 설계한다. 4장에서는 설계된 비디오 검색 모델을 계승, 캡슐화, 함수 중복등 객체 지향 특성을 이용하여 객체-관계 DBMS에 통합하고, 객체-관계 DBMS인 Informix Universal Server에서 제공하는 기능을 통해 구현한 모델에 대해 서술하도록 한다.

마지막으로 5장에서는 본 논문의 결과를 정리하고 앞으로의 연구 방향을 제시한다.

2. 관련 기술 및 연구 동향

2.1 기존의 비디오 데이터 시간 모델링

그 동안의 비디오 데이터의 시간 모델링(temporal modeling)을 살펴 보면, 비디오 데이터와 오디오 데이터의 동기화에 치우쳐 있었다[11,12]. 그러나 최근 객체-지향 DBMS와 객체-관계 DBMS에서 지원하는 blob 데이터 타입은 바이너리 데이터에 임의 접근하여 원하는 부분을 읽어 들일 수 있는 기능을 제공해 줌으로써, 비디오 데이터 자체를 시간적 단위로 분할하는 비디오 데이터 모델링이 연구되고 있다.

이러한 모델에서는 비디오 데이터를 여러 장면의 연속이라고 규정하고, 이를 개념적인 시간 간격의 집합으로 나누고, 이러한 시간적 관계를 개별적으로 또는 그룹화하여 접근하였다. 하나의 장면(shot)은 논리적으로 연관된 프레임(frame)의 연속으로 그룹화하고, 이것은 더 이상 분할되지 않는 것으로 규정하였다. 이러한 시간 관계는 데이터 요소 i를 위한 시작시간(π_i), 지속 시간(τ_i), 그리고 종료 시간(γ_i)으로 이루어 진다.

이러한 모델링하에 다음과 같은 시간 함수를 정의한다. 이러한 함수는 SQL에서 직접 사용하도록 하고, 각 함수의 리턴 값은 **true**, **false**이다.

> **before**(α,β), **after**(α,β), **overlap**(α,β),
> **during**(α,β), **equals**(α,β)....
> if $\alpha = [\pi_i, \gamma_i]$, $\beta = [\pi_j, \gamma_j]$

그림 1. 제한적인 시간 함수

따라서 위와 같이 설계된 모델링과 시간 함수에 대해, 비디오 데이터의 시간적 관계에 대한 질의어는 두 개의 논리적 분할 단위가 어떠한 시간 관계를 가지느냐에 대한 질의문밖에 수행할 수 없다. 또한, 본 논문에서 정의한 비디오 데이터내에 포함되어 있는 비디오 객체에 대한 시간관계는 전혀 고려하지 않았다.

한 예로 Informix Universal server[15,16]의 video datablade module에서는 각각의 논리적 단위를 비디오 데이터의 상대적 시작 시간과 끝시간으로 표현한다. 따라서 자신이 알고 있는 비디오 데이터의 시작 시간과 끝 시간을 기준으로 각각의 비디오 데이터의 논리적 단위가 시간 함수에 대해 어떠한 관계를 갖고 있느냐에 따라 True/False값을 리턴하는 시간 질의어만을 실행할 수 있다. 다음 그림 2는 Informix Universal Server의 Video datablade module에서 시간 질의어를 수행한 예이다.

< 질의문>
select id, **after**('10-00:00:00|20-00:00:00') **from** mediachunktest;

< 결과값 >

id	expression
1	f
2	f
3	t

그림 2. Informix Universal Server에서의 시간 질의어와 그 결과값의 예

2.2 콜럼비아 대학의 내용기반 그래픽 질의 검색 연구

MPEG-4의 등장과 함께, 비디오 데이터 내에 존재하는 비디오 객체에 대한 검색 연구가 활발히 이루어지고 있다[1,2,7,8,9]. 이러한 연구의 대부분은 비디오 객체의 화면상 위치를 검색하는 공간 질의에 치우쳐 있으며, 하나의 비디오 데이터에서만 정의 가능하고, 대수(algebra)로만 정의한 형태로 질의어로 확장되지 못한 단점을 지니고 있다. 그 중 콜롬비아 대학의 연구를 살펴 보면, 이 연구의 목적은 비디오 객체로부터 시각적 특징을 추출하고, 그것을 빠르고 효율적으로 검색하기 위한 효율적인 데이터 구조를 만들고, 이를 지원하는 내용기반 그래픽 질의어(content-based visual query)를 지원하는 것이다. 즉 비디오 데이터에서 색깔, 움직임, 질감, 모양과 같은 특성들이 동일한 연속적인 픽셀들의 집합을 궤도(trajectory)와 그들간의 공간관계를 계산하여, 움직이는 비디오 객체로 그룹화를 시켜 비디오 객체를 추출하는 것이다. 이러한 방식을 이용하여, 콜롬비아 대학에서는 VideoQ라는 시스템을 구현하였다[9,10,12]. 이 시스템은 예제를 통한 질의와 visual sketch를 통해 움직이는 하나의 비디오 객체가 시간의 흐름에 따라 공간

내에서 어떻게 이동하는가에 관한 관계를 나타내는 공간적 질의를 지원한다. 이때 비디오 객체의 특성은 {attribute, value}의 쌍(pair)으로 관계형 데이터베이스에 저장된다. 이러한 VideoQ 시스템은 비디오 데이터내의 객체에 대한 검색은 고려하였지만, 비디오 데이터의 논리적 계층구조에 대한 검색은 고려하지 않았다. 또한 그들이 지원하는 그래픽 질의는 비디오 객체에 대한 공간 질의와, 픽셀의 집합에 대한 한정된 검색에는 적당하지만, 움직이는 비디오 객체에 대한 검색에는 적당하지 않다. 또한 비디오 검색에 있어서 데이터베이스 구조가 중요하지만, 현재 구현에서는 전혀 중요하게 다루지 않고 있다.

최근에 객체 지향 방법을 기반으로 멀티미디어 데이타를 모델링하고, SQL 언어를 확장하여 시공간적 특성을 조건으로 주고 검색하는 연구가 진행되고 있다[18]. 그러나 비디오내의 객체들간의 의미적 시간적 특성을 기반으로 한 검색 모델에 대한 연구는 거의 없는 실정이다.

3. 비디오 데이터에 대한 시간 관계성

3.1 비디오 데이터베이스 구현 과정

비디오 데이터베이스의 구현 과정은 그림 3에 나타난 처리 과정을 거치게 된다. 본 논문에서 의미하는 비디오 객체(object)는 비디오 프레임에서 의미있는 물리적 대상을 의미한다. 예를 들어 사람, 집, 차 등이 여기에 속한다. 본 논문은 비디오 데이터의 논리적 계층 구조에 따른 시간 관계성을 제시한다. 또한 비디오내에 나타난 모든 객체와 객체들의 동작에 대한 분석이 완료되어 추출되었다고 가정하고, 객체와

그림 3. 시간 관계성을 기반으로 한 비디오 데이터베이스 구현 과정

동작에 대한 정보를 어떻게 데이타베이스에 표현하여 이를 기반으로 내용 기반 시간 질의(temporal content-based query)를 지원하는 데 목적을 둔다. 즉 본 연구에서는 그림 3에서 그림자 진 부분, 특히 비디오 객체들 간의 관계성과 동일한 비디오 객체가 한 비디오 내에서 가지는 관계성을 이용한 검색 기능을 제공하는 것을 주로 연구하였다.

비디오 데이터베이스 시스템에 있어, 비디오 데이터를 다루는 데는 특별한 요구가 필요하다. 비디오 데이터베이스 구현 과정을 살펴보면 다음과 같다. 원시 비디오 데이터는 비디오와 이미지 프로세싱 기술을 사용하는 비디오 분석기에 의해 비디오 객체와 구조가 인식된다. 비디오 분석기의 주된 기능은 비디오 데이터의 특성을 추출하는 것이다. 비디오 구조화 모듈은 비디오 분석기에 의해 얻은 결과로부터 사용자가 효율적으로 비디오에 접근할 수 있도록 비디오 데이터를 논리적으로 구조화하고, 인덱스를 생성한다. 이러한 구조화는 사용자가 데이터베이스 시스템에 대해 효율적으로 질의할 수 있도록 도와준다. 이때 객체 지향 모델을 사용함으로써, 비디오 데이터의 논리적 구조 표현을 좀 더 강력하게 표현해 준다. 사용자는 질의 모델을 기반으로 한 질의어를 사용하고, 이러한 과정 동안 응용 프로그램 도메인에 따른 배경 지식을 기반으로 비디오 분석, 데이터 구조화, 질의어 설계가 이루어져야 한다[14,15].

즉 그림 3에서 볼 수 있듯이 비디오 데이터베이스를 구축하기 위해서는 크게 다음과 같은 두 분야의 연구가 이루어져야 한다.

- 비디오 데이터를 논리적으로 분할하는 파싱 방법과, 특성을 자동으로 추출하는 연구
- 비디오 데이터에 대해 멀티미디어 정보를 의미적으로 모델링하고, 메타 데이터 구성과 질의어에 대한 연구

본 논문에서는 비디오 데이터베이스 구축 과정 중 두 번째 분야에 중심을 두었다. 즉 객체 지향 기술과 비디오 데이터가 가지는 시간 관계성에 기반하여, 원시 데이터(raw data)를 비디오 의미(semantics)로 파악하고, 이를 효율적인 내부표현으로 구조화하기 위한 비디오 데이터 모델을 설계하고, 이에 따른 다양한 시간 함수를 제공해 줌으로써 시간 질의어 확장에 초점을 맞추었다.

3.2 비디오 데이터의 논리적 계층 구조

비디오 자체는 원시 데이터이지만 대부분의 경우 비디오 전체를 저장하고, 검색하는 대신 비디오의 각 장면을 분할해서 저장하고 원하는 형태의 장면만을 검색할 수 있는 기능을 요구한다. 비디오 정보에 대한 초기 연구는 비디오 데이터를 물리적인 세그먼트, 즉 연속적인 프레임 주기로 분할하는 것이였으나, 최근에는 논리적인 세그먼트 단위로 구조화하려는 방법들이 연구되고 있다. 따라서 저장 공간의 효율적 사용과 시스템의 유동성을 높이기 위해 비디오 데이터의 특정부분을 논리적인 범위와 물리적으로 독립된 형태로 분리한다. 논리적인 범위는 의미기술의 기본단위로 사용하며 공유 결합되어 보다 큰 범위의 의미기술 단위로 확장되거나 새로운 프리젠테이션을 구성할 수 있다.

비디오 데이터의 시간적 의미를 구체화시키기 위해, 시간적 내포관계와, 비디오 데이터의 내용에 따라 그림 4와 같이 논리적으로 분할한 논리적 계층 구조를 갖는다고 정의한다. 비디오 데이터의 하위 레벨은 상위 레벨의 포함 관계(part_of)의 관계를 가지는 계층구조로 표현되어진다. 즉 비디오 데이터는 데이터가 담고 있는 내용에 따라 분할되어지고, 계층적 구조에 의해 정렬되어진다. 모든 노드들은 왼쪽으로부터 오른쪽으로 그들의 시간 간격에 의해 순서화 되어진다. 이러한 논리적 분할은, 비디오 데이터를 효율적으로 저장 및 공유하고, 데이터베이스로부터 효율적으로 검색 가능하게 한다.
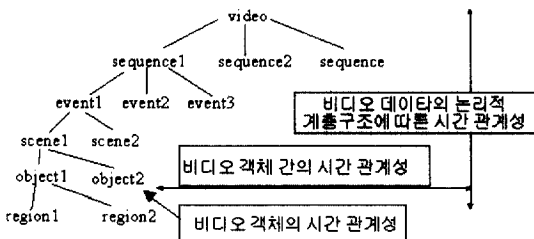
그림 4. 비디오 데이터의 계층적 구조와 각 계층에 따른 시간 관계성

비디오 데이터를 시간적 구조로 모델화하는 가장 자연스러운 방법은 시간의 흐름에 따라, 그리고 비디오 데이터의 내용에 따라 효율적으로 구조화하는 것이다. 따라서 비디오 객체에 대한 논리적 계층 구조를 정의하기 위해, 시간 모델의 특성을 나타내는 구성요소에 따라 표 1과 같은 용어를 정의한다. 우선 비디오 데이터를 시간의 간격, interval $[T_s, T_f]$로 논리적으로 분할하며, 이러한 각 논리적 분할은 비디오 데이터의 연속된 프레임의 한 일부분으로 인식된다. T_s는 주어진 비디오에서 상대적인 시작 시간(start time)을 의미하며, T_f는 주어진 비디오에서 상대적인 끝나는 시간(finish time)을 의미한다.

표 1. 비디오 데이터의 논리적 계층 구조에 따른 용어 정의

논리적 분할단위	정의
Video	하나의 비디오 데이터. 비디오 데이터 $V_i=[T_{s_i}, T_{f_i}]$로 표현된다.
Sequence	기승전결의 시간의 흐름으로 나눈 논리적 단위. 각각의 sequence $SE_i=[T_{s_i}, T_{f_i}]$로 표현되며, 하나의 비디오는 여러 개의 sequence의 합집합으로 표현되어진다. $V_i = SE_i \cup SE_j \cup \ldots\ldots SE_z$
Event	특정한 사건이 일어난 시간의 흐름으로 나눈 논리적 단위. 각각의 event $E_i=[T_{s_i}, T_{f_i}]$로 표현되어지고, 하나의 sequence는 여러 개의 event의 합집합으로 표현되어진다. $SE_i = E_i \cup E_j \cup \ldots\ldots E_z$
Scene	하나의 장면이 바뀌는 시간의 흐름으로 나눈 논리적 단위. 각각의 Scene $S_i=[T_{s_i}, T_{f_i}]$로 표현되어지고, 하나의 event는 여러 개의 scene의 합집합으로 표현되어진다. $E_i = S_i \cup S_j \cup \ldots\ldots S_z$
Object	여러 비트 스트림에 걸쳐 어떤 기준 아래로 같이 묶여지는 비디오 region의 집합. 즉 하나로 접근 가능한 의미있는 비디오 객체를 나타낸다. 예를 들어 걸어가는 사람, 달리는 자동차등이 여기에 속한다. 한 Scene S_i는 여러 객체로 구성되어진다. $Object(S_i) = \{Object_1, Object_2, \ldots\ldots, Object_k\}$
Region	색깔, 움직임, 모양, 질감과 같은 특성이 동일한 연속적인 픽셀의 집합. 하나의 비디오 객체 Object는 여러 비디오 region R_i로 이루어진다. $Region(Object) = \{R_i, R_j, \ldots\ldots, R_k\}$

3.3 비디오 데이터의 시간 관계성

그림 4에서 보이는 비디오 데이터의 구조에 따라 다음의 3가지 시간 관계성을 정의한다.

3.3.1 비디오 데이터의 논리적 계층 구조에 따른 시간 관계성

비디오 데이터는 시간 속성을 지니는 데이터의 스트림으로 표현되어진다. 위의 논리적 계층 구조에 따라 각각의 레벨의 기본 단위는, 비디오 데이터를 내용에 따라 분할한 시간상에 표현되는 하나의 사건이다. 각 계층간에, 즉 각각의 sequence, event, scene들은 상호간에 이전, 동시, 이후, 포함 등의 관계로 설정될 수 있다. 이에 따라, 사용자가 원하는 비디오 데이터에 대해 다양하게 접근할 수 있도록 제공해 주어야 한다.

3.3.2 비디오 객체들간의 시간 관계성

이러한 객체들간의 시간 관계성은 Time-Line model을 기반으로 정의한다. Time-Line 모델은 모든 사건들을 하나의 절대적인 시간 축 상에서 표현한다. 이때 사건이란 시간축 상에서 각 비디오 객체들이 상연되어지는 시작 시점과 끝시점을 의미한다.

이러한 비디오 객체간의 시간 관계성은 크게 2가지로 고려될 수 있다.

- 비디오 객체와 객체와의 시간 관계성
- 움직이는 비디오 객체와 객체사이의 시간 관계성
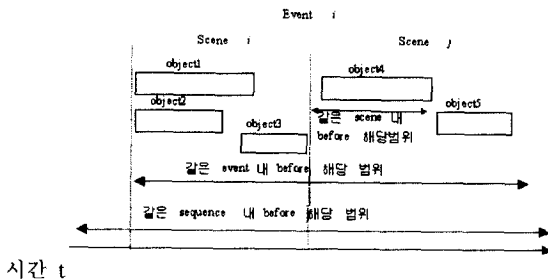
그림 5. 비디오 객체간의 시간 관계성

비디오 객체와 객체 사이의 시간 관계성을 살펴보면, 각각의 비디오 객체는 절대적인 시간 축 상에서, object = [$T_{s,} T_f$] 로 표현가능하며, 비디오 데이터 내의 여러 객체들은 시간의 흐름 하에 어떤 객체를 기점으로 하여, 이전, 동시, 이후 포함등의 관계를 갖게 된다. 또한 이러한 객체들 간의 관계속에서 찾고자 하는 논리적 분할 범위를 지정할 수 있다. 예를 들어, 그림 5에서 Object5를 중심으로 before의 관계를 갖는 객체를 검색하고자 한다면, 같은 *Scene$_j$* 내에서 전에 등장한 객체로는 Object4가 검색되어지고, 같은 *Event$_i$* 내에서 전에 등장한 객체는 Object1, Object2, Object3, Object4가 검색되어진다. 만약 같은 sequence내에서 전에 등장한 객체들을 검색하고자 한다면 더 많은 객체들이 검색되어질 것이다.

그 동안의 연구를 살펴 보면, 움직이는 비디오 객체와 객체 사이의 action에 따른 시간 관계성을 고려한 연구는 전혀 이루어지지 않았다. 따라서 여기에 대한 모델링이 필요하다. 여기에 대한 모델링은 다음 움직이는 비디오 객체에 대한 시간 관계성을 고려할 때 함께 살펴보도록 한다.

3.3.3 움직이는 비디오 객체에 대한 시간 관계성

정지 영상과 비디오 사이의 큰 차이점은 움직임과 변화성이다. 이러한 동적 객체를 시간적 지식(temporal knowledge)을 요구하여 검색하는 것 자체가 내용기반 질의의 일부가 된다. 따라서 동적 객체에 대해 모델링하는 것은 비디오 데이터베이스에 있어서 매우 흥미있는 주제가 되고 있다. 기존의 연구들을 살펴보면, 일정 시간 동안 하나의 움직이는 객체의 궤도(trajectory)를 표현함으로써, 상대적인 시간 관계를 표현하였다. 그러나 동적 객체의 움직임을 시각 질의어로 지원하는 데는 한계가 있다. 따라서 좀 더 추상화되어 모델링 할 필요성이 있다. 동적 객체의 추상화 단계를 그림으로 표현하면 다음과 같다.
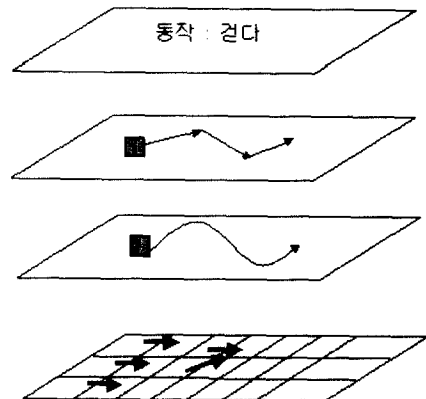
그림 6. 움직이는 비디오 객체의 동작에 대한 추상화 단계

본 논문에서는 시간적 의미를 개념적 시간 객체(conceptual temporal object)에 의해 부여하고, 어떠한 비디오 객체에 대해, 예를 들면 걸어간다거나 뛰어간다와 같은 동작(action)을 포착하여 모델링하였다. 따라서 비디오 애플리케이션의 도메인에 따라 동적 객체에 대한 기본적인 action set을 제시하고, 이에 따라 모델링 할 것을 제안한다. 이러한 action set을 기반으로 하여, 움직이는 비디오 객체의 시간 관계성은 한 scene내에서 각 객체의 action과, 객체가 action을 수행한 시작 시간(T_s)과, 끝나는 시간(T_f)의 tuple들의 list형태로 표현한다.

$$L = < (action1,\ [T_{s_1}, T_{f_1}]), (action2,\ [T_{s_2}, T_{f_2}]), \ldots, (actionN,\ [T_{s_N}, T_{f_N}]) >$$
$$If\ T_{s_I} < T_{s_{I+1}}\ \&\ T_{f_I} < T_{f_{I+1}}\ (1 <= I <= N-1)$$
$$and\ action1, action2, \ldots\ actionN \in \{ action\ set \}$$

이렇게 정의된 움직이는 비디오 객체의 시간 관계성에 따라 , 움직이는 비디오 객체와 객체사이의 시간 관계성을 정의한다. 다음 두 움직이는 비디오 객체의 action list를 다음과 같이 정의한다.

$$Object_j \text{의 } action\ list\ L1 = < (action_j1,\ [T_{s_{j1}}, T_{f_{j1}}]), (action_j2,\ [T_{s_{j2}}, T_{f_{j2}}]), \ldots, (action_jN,\ [T_{s_{jN}}, T_{f_{jN}}]) >$$
$$Object_k \text{의 } action\ list\ L2 = < (action_k1,\ [T_{s_{k1}}, T_{f_{k1}}]), (action_k2,\ [T_{s_{k2}}, T_{f_{k2}}]), \ldots, (action_kN,\ [T_{s_{kN}}, T_{f_{kN}}]) >$$

움직이는 두 비디오 객체는 각각 action을 취하게 되며, 이러한 action과 action간에는 서로 시간 관계성을 지니게 된다. 예를 들어 $Object_j$가 $action_j2$ 를 행하기 전에 $action_k2$를 취하는 비디오 객체를 찾고자 할 때, $T_{f_{k2}} < T_{s_{j2}}$가 만족된다면 $Object_k$ 가 검색되어진다.

3.4 비디오 데이터의 시간 관계성에 따른 시간 함수 설계

본 논문에서 제시하는 시간 함수는, 기존의[17] 두 객체가 어떠한 시간 관계를 가지는 가에 대한 true/flase 값을 리턴하는 제한적인 함수가 아닌, 하나의 객체를 중심으로 시간적 관계를 갖는 객체들을 리턴받는 좀 더 확장된 형태이다.

앞에서 제시한 비디오 데이터의 논리적 계층 구조에 따라 다음 3가지의 시간 함수를 설계할 수 있다.

- θ : 각 비디오의 논리적 분할에 따른 시간 함수
- δ : 각각의 비디오 객체들간의 시간 함수
- ρ : 변화하는 하나의 비디오 객체에 대한 시간 함수
- μ_i, μ_j, μ_k : 시간 함수를 적용하기 위한 파라미터 집합
- Vo : 시간 함수가 적용되어 리턴되는 객체의 집합이나 비디오 데이터의 논리적 분할 단위
- Vo = θ (μ_i)
- Vo = δ (μ_j)
- Vo = ρ (μ_k)

  if θ, δ, ρ ∈{ equal, before, before_all, after, contains, overlap, during}

확장된 시간 함수의 종류에 따른 의미는 다음과 같다.

α를 기준이 되는 데이터라 정의하자. 이 때, α는 앞에서 명시한 비디오 데이터의 논리적 분할 단위가 될 수도 있으며, 비디오 데이터 내에 존재하는 의미있는 비디오 객체도 될 수 있다. 즉 어떤 형태이든 비디오 데이터 내에서 시간의 흐름에 따라 $\alpha = [T_{s_\alpha}, T_{f_\alpha}]$ 로 표현가능하다. 그리고 시간 함수에 의해 적용되어 리턴되어지는 비디오 데이터를 β라 규정했을 때, β또한 비디오 데이터의 논리적 분할 단위이거나 비디오 객체일 수 있으며 $\beta = [T_{s_\beta}, T_{f_\beta}]$ 로 표현가능하다. γ는 비디오 데이터의 논리적 계층구조에서 α의 한 단계 상위 레벨을 나타낸다.

이러한 시간 함수 설계는 객체 지향 개념의 다형성(polymorphism)을 적용하여, 같은 이름을 가지는 시간 함수지만 다른 입력 파라미터 타입을 가지는 다양한 함수로 정의된다. 이런 경우 사용자는 잠정적으로 하나의 함수를 가지고 있기 때문에, 단지 예상되는 작업을 수행할 함수만을 생각하면 되기 때문에, 사용자에게 좀 더 편리하고 단일한 인터페이스를 제공해 줄 수 있다.

4. 시스템 구현

본 장에서는 논문에서 제안한 시간 관계성을 기반

으로 설계된 비디오 데이터 모델과 시간 함수를 기반으로 하는 시스템 구조의 효율성을 검증하기 위해 구현한 프로토타입 시스템에 대해서 기술하고, 이 시스템을 기반으로 객체-관계 DBMS의 비디오 데이터를 웹 시스템에서 제공해 주는 예를 보인다.

시스템 구현 환경은 표 2와 같다.

표 2. 객체-관계 DBMS 기반 시스템 구현 환경

서버측 운영 체제	Sun 사의Solaris 2.6
객체-관계 DBMS	Informix Universal Server V9
웹 서버	Solaris 2.6 기반의 apache_1.3.1
웹 클라이언트	웹 브라우저
개발 도구 및 언어	Informix Web Datablade Module 3.32 Informix ESQL/C Informix Datablade API Informix SPL function

4.1 시스템 구조 및 질의 처리 흐름도

시스템은 웹 서버와 데이타베이스 서버를 사용하는 3계층 클라이언트/서버 구조를 갖는다. 구현한 프로토타입 시스템의 구조는 그림 7과 같다. 시스템의 전체 구조는 물리적으로는 사용자가 검색하고자 하는 질의 인터페이스와 질의에 대한 결과를 제공하는 웹 클라이언트, 클라이언트의 요청을 받아들이는 웹 서버 그리고 비디오 데이터를 저장하고 있는 데이터베이스 서버의 세 계층으로 구성되어 있다.
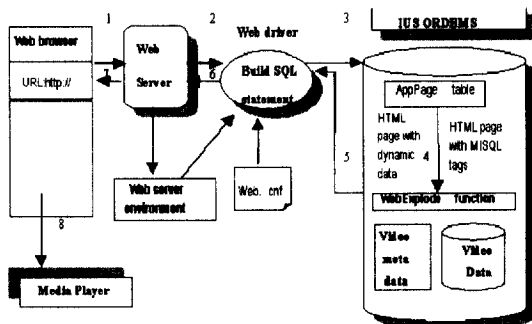
그림 7. 시스템 구조

프로토타입 시스템에 따른 질의 처리 흐름도를 간단히 살펴 보면 다음과 같다. 먼저 사용자가 시간 질의어를 서버로 보내면, 데이터베이스 내에 파싱되어 객체로써 존재하는 시간 함수를 실행시키고, 이에 따라 해당 데이터 집합을 리턴해 준다. 사용자는 해당 데이터 집합 중에서 원하는 비디오 객체를 선택하면, 비디오 데이터 처리 함수가 선택된 비디오 데이터를 사용자에게 보내주게 된다.
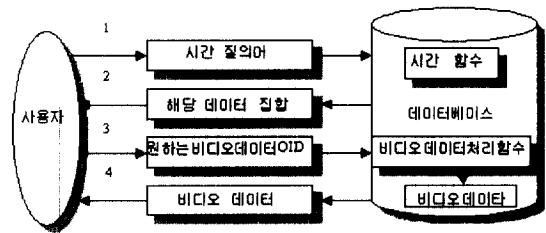
그림 8. 비디오 데이터의 시간 질의어에 따른 질의 처리 흐름도

4.2 비디오 데이터의 계층구조에 따른 데이터 타입과 테이블 정의

비디오 데이터에 대한 계층적 시간 관계성을 ORDBMS로 통합하기 위해 다음과 같이 계층적 상속 관계를 갖는 데이터 타입과 테이블을 정의한다. 이러한 타입 계층 상속의 장점은 다음과 같다.

- 데이터 모델을 모듈 방식으로 구현하기 쉽다.
- 스키마의 일관성 있는 재사용을 도와준다
- 상위 타입과 하위 타입과의 관계를 명확히 한다.
- 상위 타입의 애트리뷰트와 함수 및 연산자들을 하위 타입이 자동으로 상속받는다.
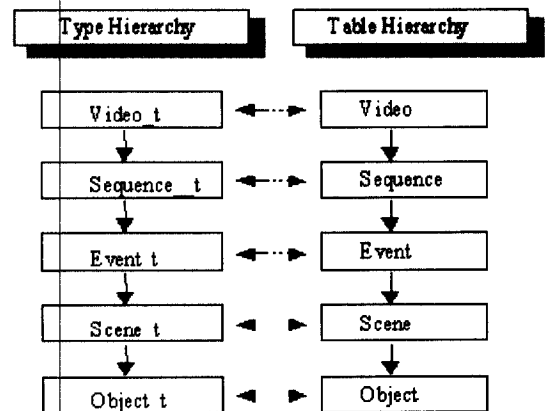
그림 9. ORDBMS에서 타입과 테이블의 상속관계

본 논문에서는 예제 도메인을 농구 비디오 데이터

```
create row type video_t (
oid          varchar(30),
  /* 각 객체를 식별하기 위한 id 값 */
video_id     integer,
  /* 비디오 데이터를 구별하기 위한 id 값 */
game_name    varchar(100),
  /* 농구 경기 이름 */
game_date    varchar(100),
  /* 농구 경기 날짜 */
description  varchar(200)
  /* 경기에 대한 묘사 */
flag         boolean);
  /* 해당되는 객체를 표현하기 위한 flag */
  /* video 데이터 타입 선언 */

create row type sequence_t ( sequence_id integer )
under video_t;

  /* video 데이터 타입을 상속받은 sequence 데이터
  타입 선언 */

create row type event_t ( event_id integer ) under
sequence_t;
  /* sequence 데이터 타입을 상속받은 event 데이터
  타입 선언 */

create row type scene_t ( scene_id integer ) under
event_t;
  /* event 데이터 타입을 상속받은 scene 데이터 타입
  선언 */

create row type object_t(
object_id integer,
object_name varchar(30),
action_list  list( action_t not null) ) under event_t;
  /* scene 데이터 타입을 상속받은 object 데이터 타입
  선언 */

create row type action_t (
action varchar(50),
s_time action_time,
f_time action_time );
  /* 비디오 객체의 action list를 나타내기 위한 타입 선
  언 */

create distinct type action_time as float;
  /* 비디오 객체가 행동하는 액션에 따른 상대적인 시
  간을 나타내는 타입 선언 */

create distinct type media_t as blob;
  /* blob 형태의 비디오 데이터를 나타내는 타입 선언 */

create table video of type video_t;
create table sequence of type sequence_t under
video;
create table event of type event_t under sequence;
create table object of type object_t under event;
```

그림 10. 비디오 데이터의 계층적 구조에 따른 데이터 타입 정의

```
create table Media_MetaData (
Oid                  varchar(30),
  /* 각 객체들이 가지고 있는 식별값 */
Vid                  integer,
  /* 실제 비디오 데이터를 가리키기 위한 식별값 */
Format_type          varchar(30),
  /* 비디오 데이터 포맷 타입 */
Capture_rate    float, /* 비디오 데이터의 캡쳐 비율 */
duration         integer,
  /* 각 객체가 나타내는 상영되는 시간 */
offset            integer);
  /* 실제 비디오 데이터에서의 옵셋 */
create table VideoMedia (
Vid         integer,
  /* 실제 비디오 데이터를 나타내는 식별값 */
bitrate     integer, /* 비디오 데이터의 비트율 */
datasize    float,  /* 비디오 데이터의 크기 */
media       media_t ); /* 실제 비디오 데이터 */
```

그림 11. 비디오 데이터를 위한 테이블 정의

로 정하여 비디오 데이터의 논리적 계층구조에 따른 타입 선언을 통하여 상속관계를 구현하였다.

객체-관계 DBMS인 Informix Universal server에서 비디오 데이터베이스 스키마는 다음과 같다. 비디오 데이터의 계층적 모델을 객체-관계 데이터베이스에서도 계층적 상속 관계로 표현하였다. 각각의 계층적 테이블은 각각 Oid 값을 가지고 있다. OidPtrMedia table에서는 이러한 Oid값과 각 Oid에 따라 offset값과 duration값을 가지고 있어서, 비디오 데이터내에서 상대적인 일부분을 가리키게 된다. 실제 비디오 데이터는 VideoMedia table에서 관리하게 된다. VideoMedia table의 Media attribute는 BLOB(Binary Large Object)형태의 비디오 데이터를 저장하는 것으로, Informix Universal Server에서는 실제 table에는 LO handle값을 가지고 있으며 실제 비디오 데이터는 sbspace라는 저장소에 저장하고 있다.

4.3 비디오 데이터의 논리적 계층구조에 따른 시간 함수

객체 관계 DBMS인 INFORMIX에서 지원하는 SPL 함수(Stored Procedure Language function)을 이용하여 다양한 시간 함수를 제공한다. SPL 함수는 데이터베이스 내 시스템 카탈로그 테이블에 객체로써 저장되어, 한번 실행되고, 저장되었을 때 더 이상 파싱될 필요없다. 따라서 클라이언트/서버 사이의 트래픽을 줄일 수 있으며, 애플리케이션의 성능을 향

상시킬 수 있다. 또한 처음에 어느 세션이 함수의 사용을 요구할 때, 데이터베이스 서버는 SPL 함수를 실행 형태로 메모리에 캐쉬하여 다른 사용자와 같이 사용하므로, 보다 효율적으로 실행될 수 있다. 또한 함수 내부 프로그램을 캡슐화함으로써, 애플리케이션은 단지 그 기능만을 생각하면 되기 때문에, 사용자에게 편리한 인터페이스를 제공 할 수 있으며, 별도의 보안을 제공해 줄 수 있다.

이러한 SPL 함수를 함수 중복(function overloading) 기능을 이용하여, 같은 이름을 가지지만, 다른 입력 파라미터 타입을 가지는 다양한 함수로 구현한다. 이것은 입력 파라미터에 기본을 두고 실행할 정확한 함수를 선택하여 실행되어진다. 이런 경우, 사용자는 잠정적으로 하나의 함수를 가지고 있기 때문에, 단지 예상되는 작업을 수행할 함수만을 생각하면 된다.

본 논문에서는 앞에서 제시한 시간 관계성과 시간 함수 설계에 따라, 움직이는 비디오 객체를 중심으로 다음과 같은 확장 시간함수를 구현하였다.

- θ : 각 비디오의 논리적 분할에 따른 시간 함수
- δ : 비디오 객체들간의 시간 함수
- ρ : 움직이는 하나의 비디오 객체에 대한 시간 함수
- Vo : 시간 함수가 적용되어 리턴되는 객체의 집합이나 비디오 데이터의 논리적 분할단위.
- Range : 사용자가 검색하고자 하는 비디오의 논리적 분할 단위
- Obj_name : 찾고자 하는 객체의 이름
- Action : 객체가 하는 행동

- Vo = θ (*Obj_name, action, range*)
- Vo = δ (*Obj_name, action1, action2, range*)
- Vo = ρ (*Obj_name, action1, action2*)
  if θ, δ, ρ ∈{ equal, before, before_all, after, contains, overlap, during}

데이터베이스 서버가 함수를 수행하고자 할 때, 데이터베이스 서버는 함수 이름과 불려진 매개변수와 일치되는 것을 찾아, 수행시킨 후 원하는 결과값을 리턴하게 된다. 본 논문에서는 예제 도메인을 농구 비디오 데이터로 정하였으므로, 비디오 객체간의 시간 함수는 움직이는 객체간의 함수만을 구현하였다.

비디오 객체 간의 시간 관계성에 따른 확장된 시간 함수의 수행과정을 간단히 살펴 보면 알고리즘 1과 같다.

```
create function temporal_function_name(
parameter_set )
returning return_set ;
① for I = 1 to c /* c : 전체 object의 갯수 */
/* 각각의 비디오 객체에 대해 다음의 조건을 실행한다.
*/
② select video_id, sequence_id, event_id
   from object
   where parameter_set ;
/* parameter 조건에 맞는 기준이 되는 비디오 객체를 찾는다.
그 객체의 video_id, sequence_id, event_id를 리턴받는다.
*/
③ if range = range_name then
/* range와 수행하고자 하는 시간 함수에 따라 video_id,
 sequence_id, event_id 값을 결정한다. */
④ update object
   set flag = 't'
   where 결정된 id값과 찾고자 하는 객체의 action에
따른 조건 ;
/* 위에서 결정된 id값들과 각 객체의 action list의 action
value,
 수행하고자 하는 시간 함수에 따라 해당되는 객체의 flag
를 모두 true로 set*/
⑤ end if;
   end for;
⑥ foreach
   select return_value into return_set
   from object
   where flag='t';
   return return_set with resume;
   end foreach;
/*  flag='t'인 객체들을 모두 검색하여 리턴한다. */
⑦ end function;
```

알고리즘 1. 비디오 객체 간의 시간 관계성에 따른 확장된 시간 함수의 수행과정

4.4 구현된 시스템을 이용한 예제

본 논문에서는 예제로 농구 비디오 데이타베이스를 사용하였다. 비디오 데이타의 시간 관계성 모델링은 비디오 응용 분야에 따라 다르게 적용된다. 농구

예제에서는 한 경기를 video로, 각 쿼터를 sequence로, 각 팀별 공격 시간을 event로 규정하여 논리적 분할을 하였다. 객체는 농구선수로 하였으며, action의 집합은 다음을 사용했다.

{'3 point shot', '2 point shot', 'holding a ball', 'slam dunk', 'steal', 'pass', 'turnover', 'block', 'free throws', 'jump a ball', dribble, foul'}

시간 함수와 응용 분야 비디오 객체들간의 시간관계성에 따른 시간 질의어의 예는 다음과 같다.

예1) 시간 질의어: '같은 event내에서 전주원 선수가 3점 슛을 하기 전에 pass한 모든 선수를 검색하라.'

실행되는 시간 함수:

Execute function **before_all**('*전주원*', *'3-pointer shot', 'pass', 'event'::range_t*);

이 질의문의 처리 결과는 action1('3-pointer shot')을 행한 객체(농구 선수)를 검색한 후, 해당되는 논리적 분할 범위(event)내에서 그 전에 action2('pass')를 행한 모든 객체들(농구선수들)을 리턴한다. 즉 한 객체의 action과 다른 객체의 action과의 시간 관계에 따라 원하는 객체를 리턴받는다.

그림 12. 비디오 객체들간의 시간 관계성에 따른 시간 질의어 사용자 입력 화면

비디오 객체의 시간관계성에 대한 시간 질의어의 예는 다음과 같다.

예2) 시간 질의어 : '드리블을 한 후에 3점 슛을 하는 모든 선수를 검색하라.'

실행되는 시간 함수 : Execute function **after** ('*', *'dribble', '3-pointer shot'*);

이 질의문의 처리결과로 객체들(운동선수)의 action list를 검색해서 dribble을 한 후 3-pointer shot을 행하는 해당 객체들을 리턴한다. 하나의 객체가 행하는 action의 시간 관계성에 따라 원하는 객체를 리턴받는다.

그림 13. 비디오 객체의 시간 관계성에 따른 시간 질의어 사용자 입력 화면

앞에서 실행된 시간 질의어 '드리블을 한 후에 3점 슛을 하는 모든 선수를 검색하라.'에 대한 실행 결과는 15개의 비디오 객체(농구 선수)가 검색되어진다. 검색되어진 비디오 객체 집합에서 보기를 원하는 비디오 객체의 OID를 입력하면 아래와 같이 Media Player에 의해 비디오 데이터가 플레이된다. 본 논문에서는 검색하기를 원하는 비디오의 논리적 단위에 따라 비디오 데이터가 재생되고, 비디오 객체의 경우는 하나의 비디오 객체가 나타난 scene을 재생한다.

5. 결론 및 향후 연구 과제

비디오 데이터의 중요한 면 중에 하나는 시간 관계성을 지닌다는 것이다. 따라서 비디오 정보에 대해 의미적으로 추상화하고, 복잡한 비디오 데이터에 대해 그들 간의 시간적 표현을 간략히 표현 가능하도록 제공해 주어야 한다. 따라서 본 논문에서는 비디오 데이터를 시간적 구조로 모델화하기 위해 시간의 흐름에 따라 또한 비디오 데이터의 내용에 따라 효율적으로 분할, 그룹화시켜 논리적 계층 구조로 표현하고, 이러한 계층구조에 따라 다음 3가지 시간 관계성을 정의하였다.
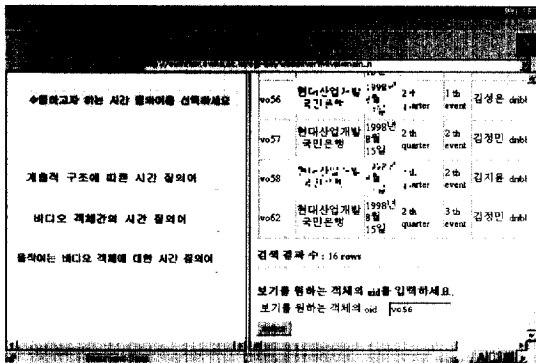
그림 14. 시간 질의어에 대한 결과 화면
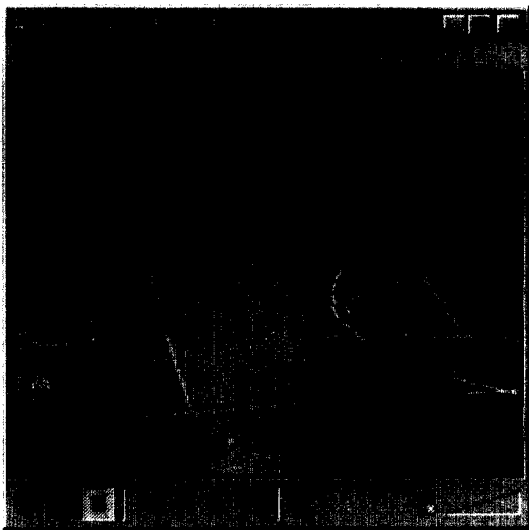
그림 15. Media Player에 의해 선택적으로 재생된 비디오 데이터 실행 화면

- 비디오 데이터의 논리적 계층 구조에 따른 시간 관계성
- 비디오 객체들간의 시간 관계성
- 움직이는 비디오 객체의 시간 관계성

이러한 비디오 데이터의 시간 관계성 모델을 객체 관계 DBMS로 통합함으로써, 비디오 구조에 따라 상속관계를 갖는 데이터 타입과 테이블 정의, 그리고 다양한 시간 함수를 제공하고 사용자에게 좀 더 다양한 선택적 재생을 가능하게 하였다.

비디오 데이터는 처리 방법이 독특하여 이들의 표현, 저장 및 검색 조건 부여 문제, 그들간의 상호 복잡한 관련성을 적절히 표현할 수 있는 모델링 문제, 이들을 사용자 의도대로 조작할 수 있는 언어의 형태 및 기능 문제, 그리고 사용자가 원하는 비디오 데이터를 용이하게 검색할 수 있는 사용자 인터페이스 문제 등이 있을 수 있다.

이에 본 논문은 기존의 연구에 비해 다음과 같은 특징을 갖는다.

- 움직이는 비디오 객체의 동작(action)을 추상화하여, 이에 대한 시간 관계성을 고려하였고, 두 움직이는 비디오 객체의 동작간의 시간 관계성을 표현하였다.
- 비디오 데이터의 3가지 시간 관계성을 모두 지원하는 확장된 시간 함수를 지원한다. 이러한 시간 함수는 하나의 객체 또는 논리적 분할 단위를 중심으로, 어떠한 시간 관계를 갖는 객체 또는 논리적 분할 단위를 리턴해 주는 형태이다.
- 확장된 시간 함수를 통하여, 비디오 데이터의 논리적 계층 구조에 따라 다양한 시간 질의어를 지원한다. 따라서 사용자가 원하는 비디오 데이터에 대해 다양하게 접근하고, 선택적으로 재생시켜 줌으로써, 비디오 데이터의 재 사용성을 높여 줄 수 있다.

향후 연구 과제로는 본 연구에서 제안한 시간 관계성에 기반한 비디오 데이터 모델을 공간적 모델과 통합하고, 움직이는 비디오 객체의 동작들을 예를 들면, 공격하는 동작과 공격 동작은 다시 여러 단계의 패스와 슛 동작으로 구성되는 등 계층적으로 모델링하여 확장된 비디오에 대한 내용 기반 검색을 제공하는 것이다.

참 고 문 헌

[1] Z. Li, M. T. Ozsu, D. Szafron, "Modeling of Moving Objects in a Video Database", *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, Ottawa, Canada, pp. 336-343, June 1997.

[2] J. Z. Li, I. A. Goralwalla, M. T. Ozsu, D. Szafron, "Modeling Video Temporal Relationships in an Object Database Management System", *IS&T/SPIE International Symposium*

on Electronic Imaging: Multimedia Computing and Networking, San Jose, USA, February 1997, pages 80-91.

[3] William I. Grosky and Ramesh Jain and Rajiv Mehrotra "The Handbook of Multimedia Information Management", Prentice Hall PTR, 1997

[4] Michael Stonebraker, "Object-Relational Database Management System," Morgan-Kanufmann, 1996

[5] Stephen W. Smoliar and HogJiang Zhang "Content-Based Video Indexing and Retrieval," *IEEE Multimedia*, pp. 66-72, Summer 1994.

[6] http://drogo.cselt.set.it/mpeg/standards/mpeg-4.htm

[7] J. Meng and S.-F. Chang, "CVEPS: A Compressed Video Editing and Parsing System," *ACM Multimedia Conference*, Boston, MA, Nov. 1996.

[8] J. Meng and S.-F. Chang, "Tools for Compressed-Domain Video Indexing and Editing," *SPIE Conference on Storage and Retrieval for Image and Video Database*, San Jose, Feb. 1996.

[9] S.-F. Chang, "Compressed-Domain Techniques for Image/Video Indexing and Manipulation," Invited Paper, *IEEE Intern. Conf. on Image Processing*, ICIP 95, Special Session on Digital Image/Video Libraries and Video-on-demand, Washington DC., Oct. 1995,

[10] S.-F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, "VideoQ- An Automatic Content-Based Video Search System Using Visual Cues," *ACM Multimedia Conference*, Nov. 1997, Seattle, WA, also Columbia University/CTR Technical Report, CTR-TR #478-97-12.

[11] 허진용 외 "MPEG-2 압축 동영상 관리 시스템에 대한 연구," 1997년도 한국정보과학회 가을 학술발표논문집, Vol. 24, No. 2, pp. 159-162.

[12] 이훈순 외 "내용 기반 검색을 위한 비디오 질의 처리기의 설계," 1998년도 한국 정보과학회 봄 학술발표논문집, Vol. 25, No.2, pp. 83-85.

[13] http://www.newmedia.com/NewMedia/96/1/Multimedia_Database.htm

[14] A. Ghafoor, "Multimedia Database Management System," *ACM Computing Surveys*, vol. 27, no. 4, pp. 593-598, Dec. 1995.

[15] Informix, Extending Informix Universal Server, Informix Press, 1997

[16] Informix, *Informix Video Foundation Datablade Module : User's Guide*, Informix Press, 1997

[17] T. D. C. Little and A. Ghafoor, "Interval-based Conceptual Models for Time-Dependent Multimedia Data," IEEE Trans. on Knowledge and Data Engineering, Vol. 5, No. 4, 1993.

[18] 이 중화 외 3인, "멀티미디어 데이터를 지원하기 위한 SQL 확장," 멀티미디어학회논문지, Vol. 2, No. 2, pp. 109-119, June 1999.

최 지 희

1993. 3～1997. 2 이화여자대학교 컴퓨터학과(공학사)
1997. 3～1999. 2 이화여자대학교 컴퓨터학과(공학석사)
1999년～현재 LG Telecom 근무
관심분야 : 멀티미디어 데이터베이스, 무선데이터 서비스

용 환 승

1983년 서울대학교 전자계산기공학과(공학사)
1985년 서울대학교 전자계산기공학과(공학석사)
1985년～1989년 한국전자통신연구소 연구원
1989년～1994년 서울대 컴퓨터공학과 (공학박사)
1994년 서울대 컴퓨터신기술공동연구소 특별연구원
1995년～현재 이화여자대학교 컴퓨터학과 조교수
관심분야 : 데이타베이스, 멀티미디어, 데이타 웨어하우징